

# Parallel DNS and LES of Turbulence and Flow-Structure Interactions

R.M. Kirby, Y. Du, D. Lucor, X. Ma,  
G-S Karamanos and G.E. Karniadakis \*

Division of Applied Mathematics  
Brown University

## Abstract

We present parallel direct (DNS) and large-eddy simulations (LES) of wall-bounded turbulence and of flow-structure interactions involving cables, beams, and flexible eels. We analyze the results by constructing low-dimensional models for these flows based on extraction of appropriate eigenfunctions from the full simulations.

The main results this year include:

- A new drag reduction technique based on traveling waves (work published in Science [1]).
- A spectral vanishing viscosity method and dealising algorithms that allow simulations at high Reynolds number.
- Visualization of vortex dislocations and vortex splits in shear flows past vibrating flexible beams.

## 1 Technical Approach

All simulations are based on the parallel code *NekTar*, which is a C++/MPI implementation of the multi-resolution spectral/*hp* element method. In the spectral/*hp* method, convergence can be obtained by either h-refinement or p-refinement. Usually a Galerkin formulation is employed but in more recent work involving compressible flows a new discontinuous Galerkin method has been followed, which we present briefly next.

### 1.1 Discontinuous Galerkin Methods - DGM

In the last few years, the PI has been supported by DoD agencies to develop new high-order algorithms for turbulence and other CFD applications as well as for MHD

---

\*Principal investigator and corresponding author

simulations. The main result of this work has been the formulation and implementation of a *discontinuous Galerkin algorithm* for hyperbolic conservation laws and for the compressible Navier-Stokes and MHD equations in two- and three-dimensions on structured and unstructured grids. We have designed DGM as a high-order finite volume method using an orthogonal spectral basis in terms of generalized Jacobi polynomials [2]. The main features of DGM are:

- It is a high-order finite volume method.
- It is flux-based and thus conservative by construction.
- It is  $L_2$ -stable (theoretical proof) and thus does not require explicit limiters.
- It is matrix-free as the trial basis is orthogonal-spectral and this leads to high efficiency.
- It uses standard Riemann solvers and this leads to robust simulations.

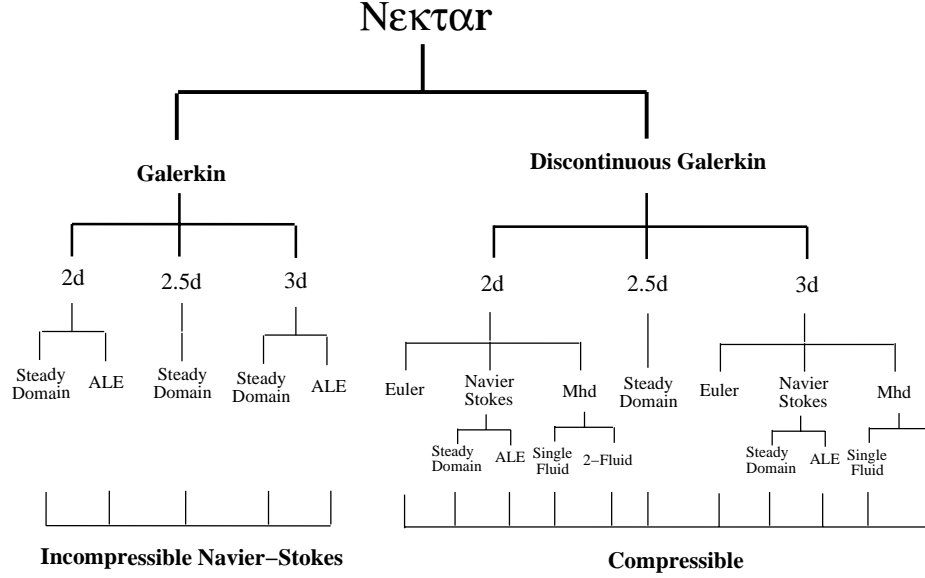


Figure 1: Hierarchy of the  $\mathcal{N}\epsilon\kappa\mathcal{T}\alpha r$  code. Note that “2.5d” refers to a three-dimensional capability with one of the directions being homogeneous in the geometry. Also, ALE refers to moving computational domains required in dynamic flow-structure interactions. The two-fluid MHD refers to a formulation for non-equilibrium plasma.

## 1.2 The $\mathcal{N}\epsilon\kappa\mathcal{T}\alpha r$ Code

The DGM algorithms have been integrated in the  $C++/MPI$  code  $\mathcal{N}\epsilon\kappa\mathcal{T}\alpha r$ , which is a general purpose parallel code for simulating incompressible, compressible and plasma flows in unsteady three-dimensional geometries. The major algorithmic

developments are described in [3] and [4] and the capabilities are summarized in figure 1. The code uses meshes similar to standard finite element and finite volume meshes, consisting of structured or unstructured grids or a combination of both. The formulation is also similar to those methods, corresponding to Galerkin and discontinuous Galerkin projections for the incompressible and compressible Navier-Stokes equations, respectively. Field variables, data and geometry are represented in terms of hierarchical (Jacobi) polynomial expansions [2]; both iso-parametric and super-parametric representations are employed. These expansions are ordered in vertex, edge, face and interior (or bubble) modes. For the Galerkin formulation, the required  $C^0$  continuity across elements is imposed by choosing appropriately the edge (and face in 3D) modes; at low order expansions this formulation reduces to the standard finite element formulation. The discontinuous Galerkin is a flux-based formulation, and all field variables have  $L^2$  continuity; at low order this formulation reduces to the standard finite volume formulation. This new generation of Galerkin and discontinuous Galerkin spectral/ $hp$  element methods implemented in the code ***NEKTAR*** do not replace but rather extend the classical finite element and finite volumes that the CFD practitioners are familiar with [2]. The additional advantages are that convergence of the discretization and thus solution verification can be obtained without remeshing (h-refinement), and that the quality of the solution does not depend on the quality of the original discretization. In figure 1 we summarize the major current capabilities of the general ***NEKTAR*** code for incompressible, compressible and plasma flows.

### 1.3 3D Visualization/Steering in Immersive Environments

We have developed new software in collaboration with IBM researchers to exploit the capability provided with our immersive facility (CAVE). Specifically, ***NEKTAR*** has been interfaced with new software that drives the CAVE's 4 walls and creates a fully immersive and fully interactive environment. This new software is called **JOT**, and it was produced in collaboration with Prof. A. van Dam's group from the Computer Science Department. In order to combine ***NEKTAR*** with JOT, ***NEKTAR*** was encapsulated into a class using the class constructs available in C++. Within the JOT software hierarchy, a volume data class was declared. Using the class inheritance properties of C++, we then created a **NEKTARdata** class which is an inherited data type of JOT's volume data class. The **NEKTARdata** class contained all the ***NEKTAR*** data structures necessary for visualizing ***NEKTAR*** field data. A collection of access methods for the **NEKTARdata** class were written so that JOT could easily access ***NEKTAR*** information while knowing nothing about ***NEKTAR***'s software structure. In preliminary work, we have tested the flow past a vibrating beam as an example of flow-structure interaction.

### 1.4 Impact and ***NEKTAR*** Distribution

The PI organized with colleagues (B. Cockburn and C.-W. Shu) the first International Conference on Discontinuous Galerkin Methods (May 1999) and edited the Proceedings in a hardbound book (Springer-Verlag), which is the first main reference on this

important topic [5]. The work sponsored by DoD agencies (F-15 electro-magnetic simulation) was selected to be on the cover of the Springer-Verlag book; also our work on DNS of flow past a flexible 3D wing was on the cover of the book on DNS-LES (June 1999) edited by D. Knight and L. Sakell (Kluwer). Work based on the MHD work led to a new mechanism of suppressing turbulence using an MHD-induced transverse traveling wave. This work is published in the May 19<sup>th</sup> (2000) issue of Science (Y. Du & G.E. Karniadakis), and a related patent is pending.

The code ***NεκTar*** has been distributed to more than one dozen Universities and Laboratories. Some of them include Kirkland AFB, Boeing, Inc., Berkeley, MIT, Caltech, Cornell University, Penn State University, CCNY, SUNY Buffalo, University of Minnesota, University of Wisconsin, Imperial College, Oxford Computing Laboratory, University of Tokyo, University of Bologna, North Carolina University, Florida State University, Norwegian University of Science & Technology, Sandia Labs, OAK Ridge Labs, Nielsen, Inc. etc. There is now documentation of the code both for users as well as developers. This is a *C++/MPI* **OpenSource code** that runs on all available platforms including clusters of PCs.

## 2 Drag Reduction Using Transverse Traveling Waves

The control of wall-bounded flows in an effort to reduce viscous stresses has been studied since Prandtl first used a trip wire to trigger transition in the boundary layer. In these early experiments, observable decreases in the shear stress at the wall were produced. Since then, a wide variety of experimental and more recently numerical studies have been performed to determine efficient and feasible control mechanisms for wall shear stress modification [6]. The development of turbulence control mechanisms and corresponding shear stress modification depends upon the observations that organised structures in the near-wall turbulence and friction drag are inevitably linked [7].

Two key elements of this view have been the discovery of low speed streaks in turbulent boundary layers as well as the verification of Theodorsen’s hypothesis on the existence of stable three-dimensional vortical structures, called hairpins, in experiments. More specifically, vortex pairs (loops), horseshoes or hairpin-like vortices have been recognised as the dominant flow structure in the near-wall region of turbulent boundary layers, and many researchers have colorfully described boundary layers as a “forest” or a “jungle” of hairpins. In figure 1 we plot the results of a direct numerical simulation (DNS) for flow over a smooth wall. We see that indeed near-wall turbulence is populated with such hairpin-like structures, and that low speed streaks are located exactly between the “legs” of these vortex structures.

Turbulence control methods have been developed under the assumption that the turbulence production cycle could be favourably altered, stabilised, or reduced in intensity by the manipulation and alteration of low-speed streaks, quasi-streamwise vortices, the viscous sublayer, or the hairpin-like structures that populate the near-wall region. To this end, the use of small grooves or riblets mounted on the wall

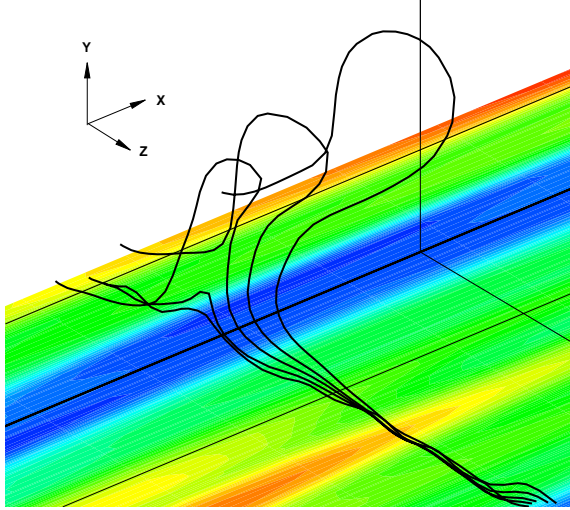


Figure 2: Hairpin-like structures and instantaneous streamwise velocity contours for turbulent flow over a smooth wall ( $Re_\tau \approx 150$ ). Blue colour indicates low-speed streaks and red colour high-speed streaks; green corresponds to average velocity values.

surface has proved to be effective in partially suppressing turbulence and reducing the drag force by about 5% to 10%. We can also imagine the use of applied near-wall forces which would act to redistribute the streamwise vorticity present in hairpin-like structures in an effort to modify the shear stress and turbulent momentum transport at the wall.

We present here results (first published in [1]) obtained from the application of such a force along the *spanwise direction* acting within the viscous sublayer and decaying exponentially away from the wall. In particular, the following travelling wave force is employed

$$F_z = I e^{-y/\Delta} \sin\left(\frac{2\pi}{\lambda_z} z - \frac{2\pi}{T} t\right) \quad (1)$$

where  $I$  is the amplitude of excitation,  $\lambda_z$  is the wavelength (along the span), and  $T$  is the time period. Here  $\Delta^+ = u_* \Delta / \nu = 5$  (in wall units), where  $u_*$  is the wall shear velocity and  $\nu$  the kinematic viscosity. The computational domain is a channel, and the force is acting on the lower wall while the upper wall is unaffected. The Reynolds number is  $Re_* \approx 150$  (based on the wall shear velocity  $u_*$  and the channel half-width). The size of the computational domain is  $L_x = 2100 \times L_y = 300 \times L_z = 840$  in wall units, in the streamwise, normal and spanwise directions, respectively. The discretisation is performed with spectral/hp element methods using the code ***NεκTαr***.

The main effect of the action of the travelling wave, when it is effective in reducing the drag force on the controlled wall, is to weaken and in many cases to completely eliminate the wall streaks. This is shown in figure 2 that plots instantaneous near-wall streamwise velocity contours both at the controlled lower wall (bottom) and at the upper uncontrolled wall (top); these planes are at  $y^+ = 4$  away from the walls. We see

that while the upper wall exhibits the familiar turbulence structure consisting of pairs of high-speed and low-speed streaks with the characteristic spacing of about 100 wall units, no such pairs can be found near the controlled surface. Instead, a wide “ribbon” of low-speed velocity is formed in the region where the streamwise vorticity achieves positive values at the same instance (plot not shown here). This is a significant new finding, in view of the fact that streaks and streak spacing are usually difficult to alter even in cases where a very large amount of drag reduction has been obtained.

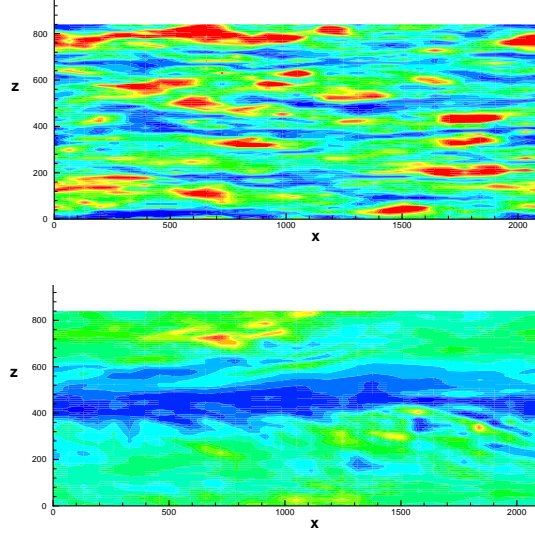


Figure 3: Instantaneous flow visualisations of wall streaks. Top: no-control; Bottom: Travelling wave excitation corresponding to  $I = 1$ ,  $\lambda_z^+ = 840$  and  $T^+ = 50$ ;  $L_z^+ = 840$ . Blue colour indicates low-speed streaks and yellow-red high-speed streaks.

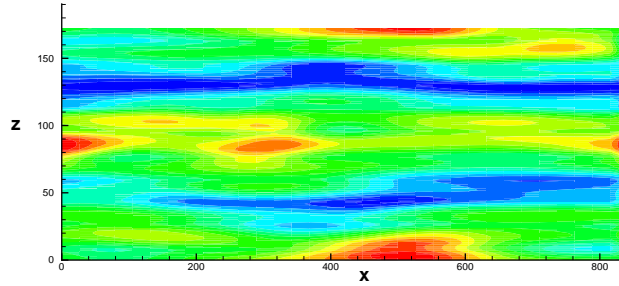


Figure 4: Instantaneous streamwise velocity contours near a wall mounted with riblets.  $L_z^+ = 200$ . Blue colour indicates low-speed streaks and yellow-red high-speed streaks.

In order to contrast this finding with the turbulence structure resulted from other drag reducing techniques, we plot in figure 3 the wall streaks visualised in simulations of turbulent flow over streamwise riblets. Although a drag reduction of 5% is achieved in this case, the lateral spacing of the streaks does not change but the streaks appear

more elongated compared to the uncontrolled case. For a different shape of riblets, the so-called “vee design”, a larger amount of drag reduction is achieved with the spacing of the streaks increased by about 10%. Similarly, if instead of a travelling wave an oscillatory flow along the spanwise direction is imposed by a force similar to the one described in equation (1) a drag reduction of 30% is achieved at the optimum period  $T^+ \approx 100$ . However, the wall streaks are still present, as shown in figure 4, but they are inclined with respect to the flow direction due to the spanwise flow component. In both cases, despite these obvious modifications the fundamental structure of near-wall streaks and hairpins remains the same unlike the travelling wave excitation, where no wall streaks or hairpin vortices can be found.

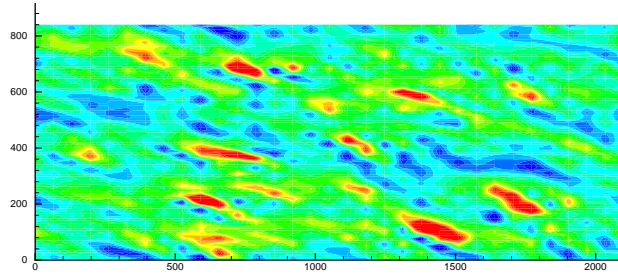


Figure 5: Streamwise velocity contours near a wall controlled by an oscillatory force along the spanwise direction;  $T^+ = 100$ .  $L_z^+ = 840$ . Blue colour indicates low-speed streaks and yellow-red high-speed streaks.

We now turn to the quantification of the corresponding drag reduction obtained from suppressing the near-wall turbulence via a travelling wave. We have found via systematic DNS experimentation that the specific waveform, i.e. the parameters  $T, \lambda_z, I$ , are critical in obtaining turbulence suppression and correspondingly drag reduction. For example, for cases with  $I = 1$ ;  $\lambda_z^+ = 840$  and  $T^+ = 100, 50$  and  $25$ , drag reduction of approximately 30% is obtained for  $T^+ = 50$ , and 15% for  $T^+ = 25$ , while the drag force is increased for  $T^+ = 100$ . This last result is rather surprising if we compare it with the oscillatory flow along the span. In the latter case, the time period  $T^+ \approx 100$  leads to largest amount of drag reduction, whereas in the travelling wave case  $T^+ = 50$  gives the maximum drag reduction among the three cases. This finding points to the fact that the underlying mechanism is fundamentally different between the two cases. Specifically, the former is related to the action of the Stokes layer while the latter can best be explained with the stabilisation of the near-wall streaks and the recent theories on regeneration mechanisms of near-wall turbulence. Another significant difference between these two cases is that in the oscillatory flow the amount of drag reduction is increased as the amplitude of excitation  $I$  increases, unlike the travelling wave excitation where this is not the case.

The finding regarding the amplitude suggests that the energetics are more favourable for the travelling wave excitation. Furthermore, we have found that it is the total amount of energy input that is critical in suppressing turbulence, and this can be expressed by the product  $I \times T^+$ . Evidence for such criterion is provided in figure

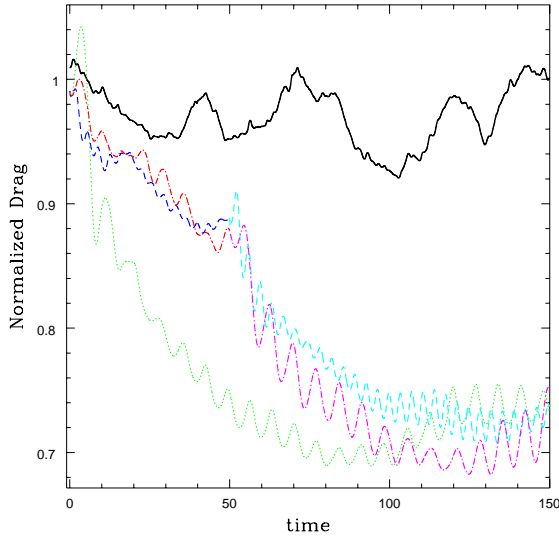


Figure 6: Drag force time-history for  $\lambda_z^+ = 840$ . Time is in convective units.

5, where we plot the time history of drag force normalised with its initial no-control value. Also, included in the plot is the normalised drag force on the upper (no-control) wall which oscillates around unity. In the plot, the curve that corresponds to  $T^+ = 50$  and  $I = 0.5$  is very close to the curve corresponding to  $T^+ = 25$  and  $I = 1$ , and both curves are for the same total energy input. To further test this hypothesis, we set  $I = 2$  for  $T^+ = 25$  and  $I = 1$  for  $T^+ = 50$  at *time* = 50, and continue the simulation. Now both cases have the same energy input as the case shown by the lowest curve in the plot, which corresponds to  $I = 1; T^+ = 50$ . We can see that there is a sudden decrease in the drag force at the beginning and eventually both curves follow each other closely towards the lowest curve in the plot. Of course, not every combination of  $(I, T^+)$  will result in drag reduction. For example, we have carried out a simulation with  $I = 0.25$  and  $T^+ = 200$  which led to a drag increase. This means that we first have to find the appropriate range of frequency, and subsequently to optimize the efficiency of this control mechanism.

In applications, it is not always possible to induce a perfect wave as the one produced in the current DNS. Moreover, from the efficiency point of view it is better to produce a travelling wave motion by discrete actuators that are sparsely distributed on the wall surface. In these cases, it is clear that an *approximate* travelling wave is produced, and the question is if such non-ideal excitation also suppresses wall turbulence. We have performed several DNS experiments to investigate this question. We found that the same result is obtained, i.e. drag reduction of about 30%, but the excitation frequency needs to be higher as the “effective” wavelength is smaller. This, in turn, suggests the importance of the *phase speed* of the travelling wave, which has to be above a certain value, e.g. at least one-third of the free-stream velocity, in order to be effective in suppressing turbulence. Another parameter that determines the net amount of drag reduction is the penetration length  $\Delta$  in equation (1). By



increasing its value, a larger amount of drag reduction is achieved but for  $\Delta$  much larger than the viscous sublayer a drag increase is obtained.

### 3 Flow-Structure Interactions

The main runs for this part include:

- Flow past long cables and beams subject to vortex-induced-vibrations (VIV) (see figures 7 and 8), and
- Flow past nonlinear cables and other nonlinear structures, (see figure 9).

The work under the previous grant has been published in [8, 9, 10, 11, 12] and [13, 14, 10, 15, 16, 17, 18], and has supported the Ph.D. theses of Evangelinos (1999), Kirby and Lucor (in progress). More specifically, the following algorithmic and simulation milestones were achieved.

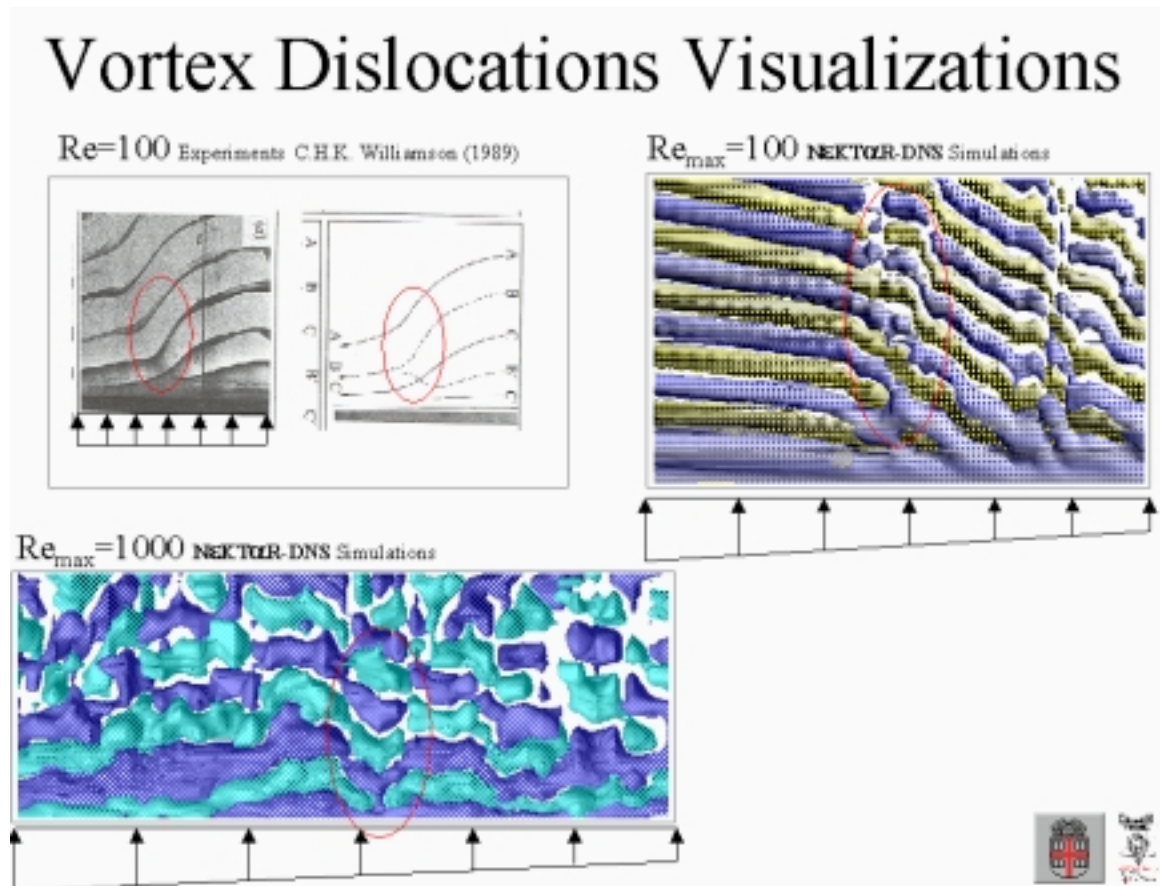


Figure 7: Comparison of DNS with flow visualizations of Williamson.

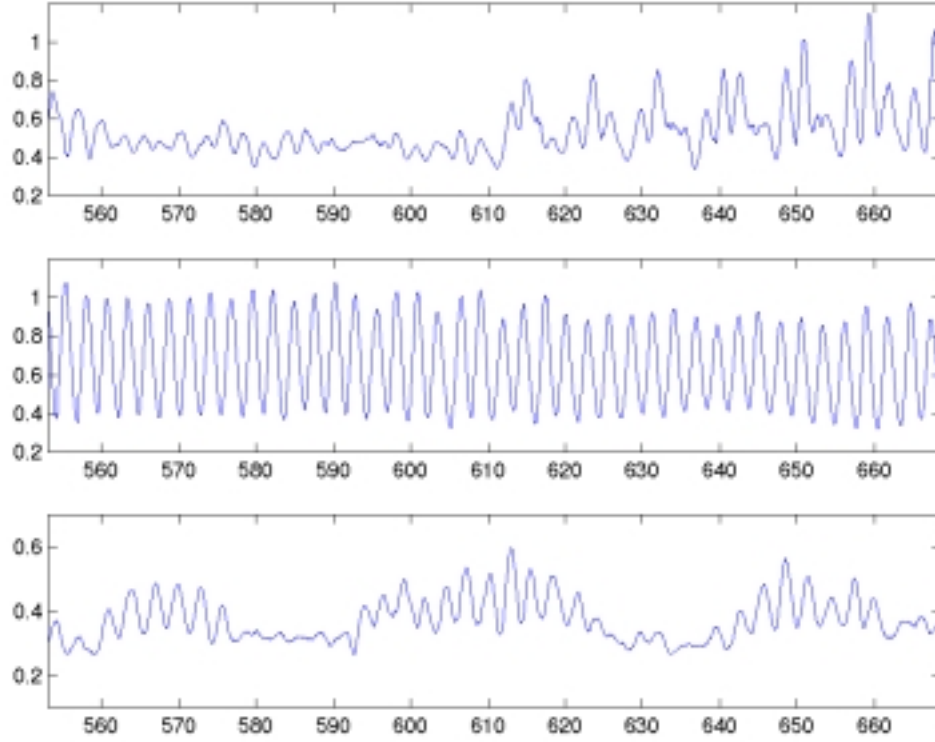


Figure 8: Effect of vortex dislocations on the drag coefficient. The middle plot is the sectional  $C_D$  away from the dislocations and the other two plots correspond to different intensity dislocations along the span.  $Re = 1,000$ .

### 3.1 Shear Inflow - Linear Structures

We have simulated VIV of cables in the turbulent regime subject to very strong shear inflow for cables up to 1000 cable diameters. This is the first time that such realistic conditions both for laboratory experiments as well as for field experiments are simulated.

In the first case, we made comparisons with the experiments of Williamson on vortex dislocation (see figure 7) and with the experiments of Trintafyllou on vortex splits. For field experiments, we have made comparisons with data of Chevron and data of Norsk Hydro. A new finding that has not been observed before in either laboratory or field experiments is the effect of vortex splits and dislocations on the drag coefficient. In figure 8 we plot the time histories of the drag coefficient at one station close to a vortex dislocation and at another two stations away from it. We see that the result of vortex dislocations (also vortex splits) on the structure's forces is indeed dramatic.

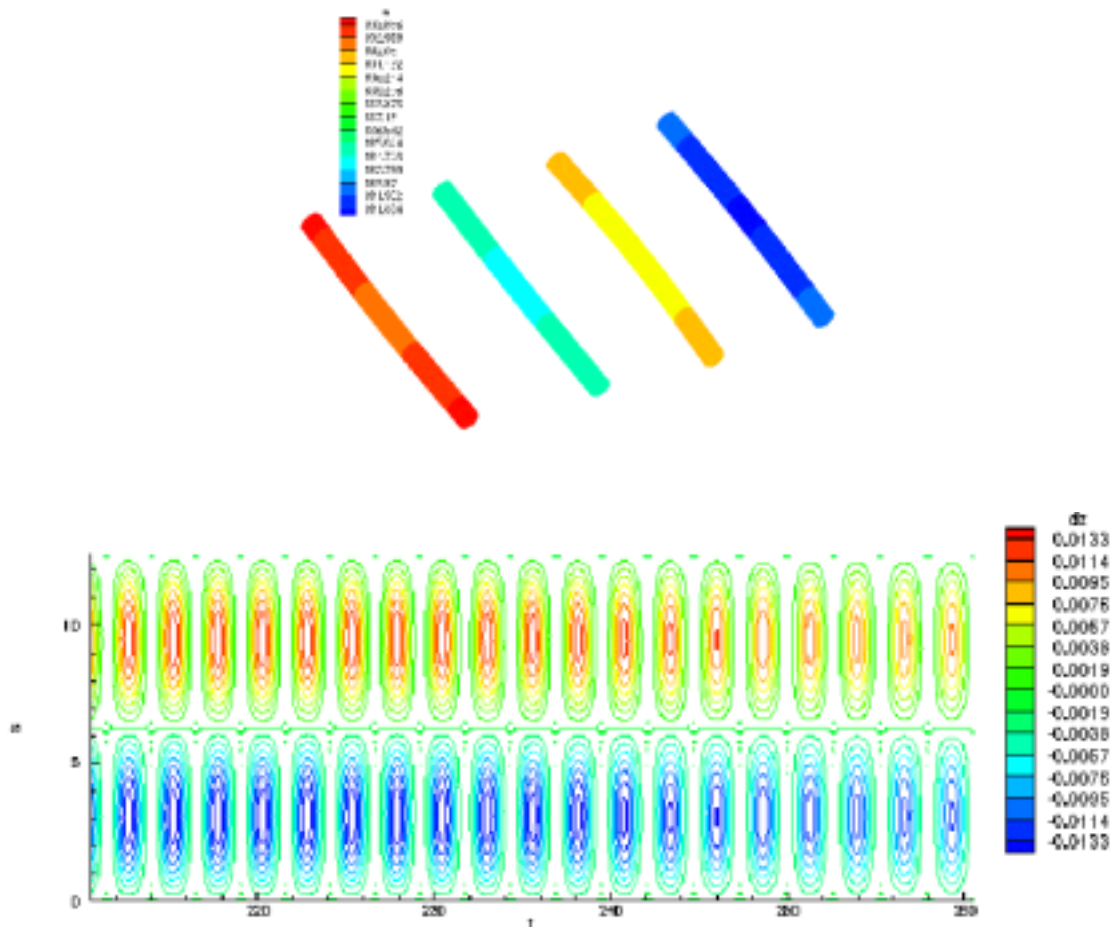


Figure 9: Nonlinear cable subject to VIV: Variable tension (top) and axial displacement oscillations.

### 3.2 Nonlinear Structures

We have developed fully nonlinear models for cables and beams and have obtained results for VIV of nonlinear cables (see figure 9). We initially used a small mesh and investigated the effect of non-zero initial conditions with small and large variations in tension. We observed that the latter lead to a breakdown of most large features in the structural response and induce large pressure oscillations in the flow. We also observed small scale temporal oscillations in the lift and spanwise forces due to the minimalist features of the mesh. In the case of the initial condition with small variations in tension we observed persistent standing wave patterns for the displacement in the cross-flow direction and the drag coefficient. The spanwise displacement (and large scale features in the spanwise force distribution) exhibited a different type of time-periodic pattern, related to a standing wave. The flow structures observed in the wake were qualitatively similar to the ones seen in simulations with linear structural models that had a standing wave response.

In order to address some of the problems with these calculations we started a calculation using a much large and more refined mesh, starting from zero initial conditions for our structure and letting the flow drive the motion. Boundary effects became quickly visible in the hydrodynamic force distributions and the structure exhibited longitudinal vibrations and increase in tension in the first 10 non-dimensional time units. These simulations exhibit a breaking of the symmetry about the mid-span, and it has been free of the pressure oscillations that appeared in the smaller domain calculations.

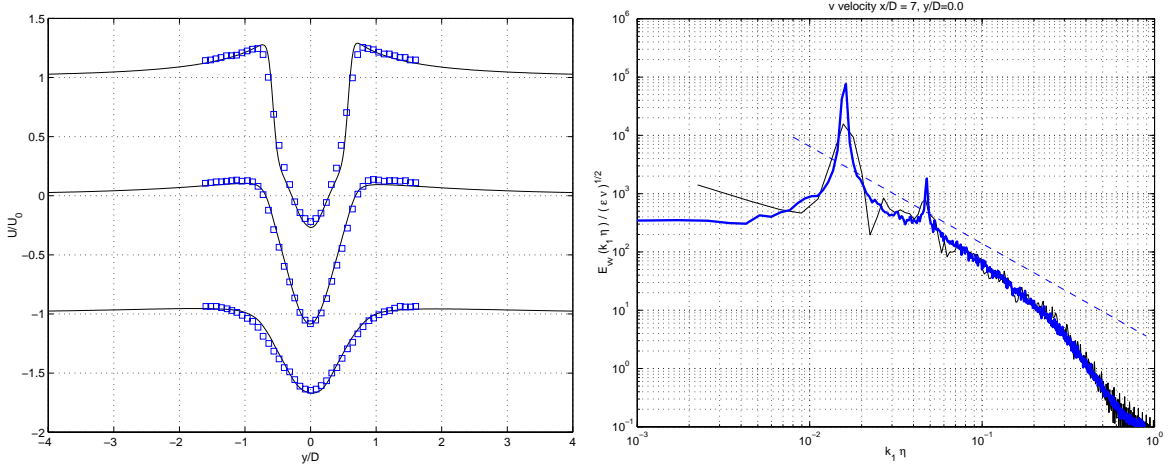


Figure 10: Velocity profiles (left) and energy spectrum (right) in the near-wake of turbulent flow past a cylinder. The 3D simulations were obtained with *NekTar* and show good agreement with the experimental data of Lurenco & Shih (1993) for velocity profiles and Ong & Wallace (1996) (thick line) for spectrum. The inertial range exhibits the Kolmogorov universal law.

### 3.3 Comparison with Experiments

We have performed several comparisons with the experiments for both moving and stationary cylinder but one that stands out, as it defines the state-of-the-art in the DNS and LES field, is flow past a cylinder in the turbulent regime. The comparison is shown in figure 10 that plots results from a three-dimensional simulation of turbulent flow past a 3D cylinder at  $Re = 3,900$ . The results are taken from [13] where more details can be found. From the spectrum plot we see that the best agreement is obtained at the small scales represented by the inertial range (i.e., high wave numbers), which are independent of the details of boundary conditions. However, there is some disagreement in the large scales as those depend on the inflow boundary conditions which in the experiment are close to uniform inflow but within some (finite) variance. This, therefore, is a case that can be simulated realistically with stochastic spectral/ $hp$  element method next year as it will include the uncertainty associated with the assumption of uniform inflow.

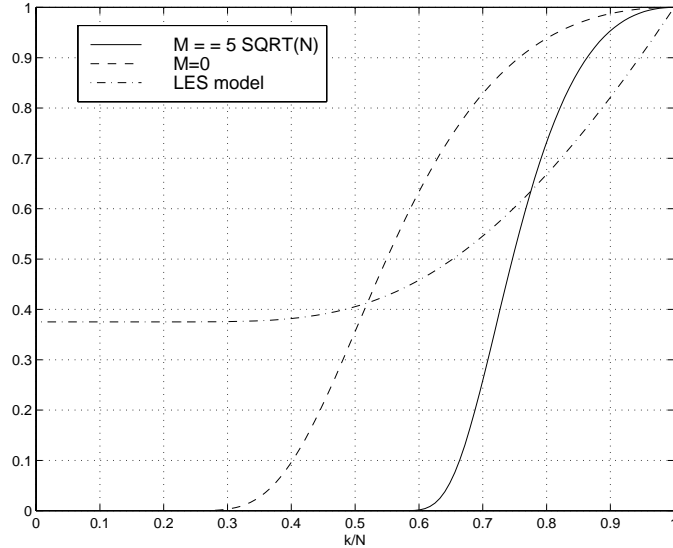


Figure 11: Normalized viscosity kernels for the spectral vanishing viscosity (dash line denotes the accuracy barrier, and solid line denotes the stability barrier) and the Kraichnan/Chollet-Lesieur viscosity (dash-dot line). Note that classical eddy-viscosity models dissipate even at the energetic scales (low modes) modifying erroneously the flow field.

### 3.4 Spectral Vanishing Viscosity: SVV-LES

An important development during the course of the previous grant was the formulation of a spectral vanishing viscosity (SVV) method. This is a new approach to control unresolved scales (high-modes) of the flow, which are explicitly constructed as the spectral/ $hp$  element method is hierarchical. In particular, a second-order viscosity *convolution operator* was constructed that is activated for the high modes unlike the classical LES approach where all modes are affected (see figure 11). This method

was motivated by the work of Tadmor [19] for nonlinear conservation laws for spectral methods. It is appropriate for incompressible flow simulations and no filtering is involved in the formulation. This method resolves the apparent conflict that exists between the formal accuracy of the numerical method and monotonicity, i.e., the implicit or explicit viscosity added in LES to control the energy of high modes. Specifically, the theory provides limits below which the spectral accuracy of the method is not preserved and above which stability is not guaranteed (see figure 11). Several benchmark simulations have been performed, e.g. incompressible turbulent channel flow at  $R_* = 180$  and  $R_* = 395$  that validate the theory [15]. One drawback of the current implementation, which we will address in the future, is that the method is not implicitly adaptive as in the standard eddy-viscosity LES models. In summary, the main features of the SVV-LES approach are:

- It is based on nonlinear theory unlike other LES approaches that lack rigorous theoretical justification.
- It is monotonicity preserving and thus it prevents erroneous oscillations.
- It maintains spectral accuracy.
- No explicit filtering is needed.
- The convolution operator is second-order and thus it is appropriate for finite-element formulations, unlike the hyper-viscosity kernels proposed by other researchers.
- It is an efficient approach as the overhead for the SVV implementation is a small fraction of what is required for the standard Smagorinsky LES implementation.

### 3.5 ALE-Graph Algorithm

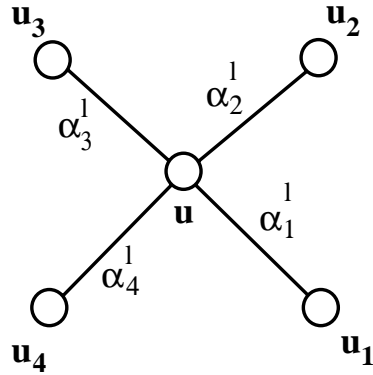


Figure 12: Graph showing vertices with associated velocities and edges with associated weights.

The grid velocity is arbitrary in the Arbitrary-Lagrangian-Eulerian (ALE) formulation we employ for flow-structure interactions. Therefore, great latitude exists in

the choice of technique for updating it. Mesh constraints such as smoothness, consistency, and lack of edge crossover, combined with computational constraints such as memory use and efficiency dictate the update algorithm used. In our work, we address the problem of solving for the mesh velocity in terms of its graph theory equivalent problem. Mesh positions are obtained using methods based on a graph theory analogy to the spring problem. Vertices are treated as *nodes*, while edges are treated as *springs* of varying length and tension. At each time step, the mesh coordinate positions are updated by equilibration of the spring network. Once the new vertex positions are calculated, the mesh velocity is obtained through differences between the original and equilibrated mesh vertex positions.

Specifically, we incorporate the idea of variable diffusivity while maintaining computational efficiency by avoiding solving full Laplacian equations. The method we use for updating the mesh velocity is a variation of the barycenter method [20] and relies on graph theory. Given the graph  $G = (V, E)$  of element vertices  $V$  and connecting edges  $E$ , we define a partition  $V = V_0 \cup V_1 \cup V_2$  of  $V$  such that  $V_0$  contains all vertices affixed to the moving boundary,  $V_1$  contains all vertices on the outer boundary of the computational domain, and  $V_2$  contains all remaining interior vertices. To create the effect of variable diffusivity, we use the *concept of layers*. As is pointed in [21], it is desirable for the vertices very close to the moving boundary to have a grid velocity almost equivalent to that of the boundary. Hence, locally the mesh appears to move with solid movement, whereas far away from the moving boundary the velocity must gradually go to zero. To accomplish this in our formulation, we use the concept of local tension within layers to allow us to prescribe the rigidity of our system. Each vertex is assigned to a layer value which heuristically denotes its distance from the moving boundary. Weights are chosen such that vertices closer to the moving boundary have a higher influence on the updated velocity value. To find the updated grid velocity  $u^g$ , at a vertex  $v \in V_2$ , we use a force-directed method. Given a configuration as in figure 12, the grid velocity at the center vertex is given by:

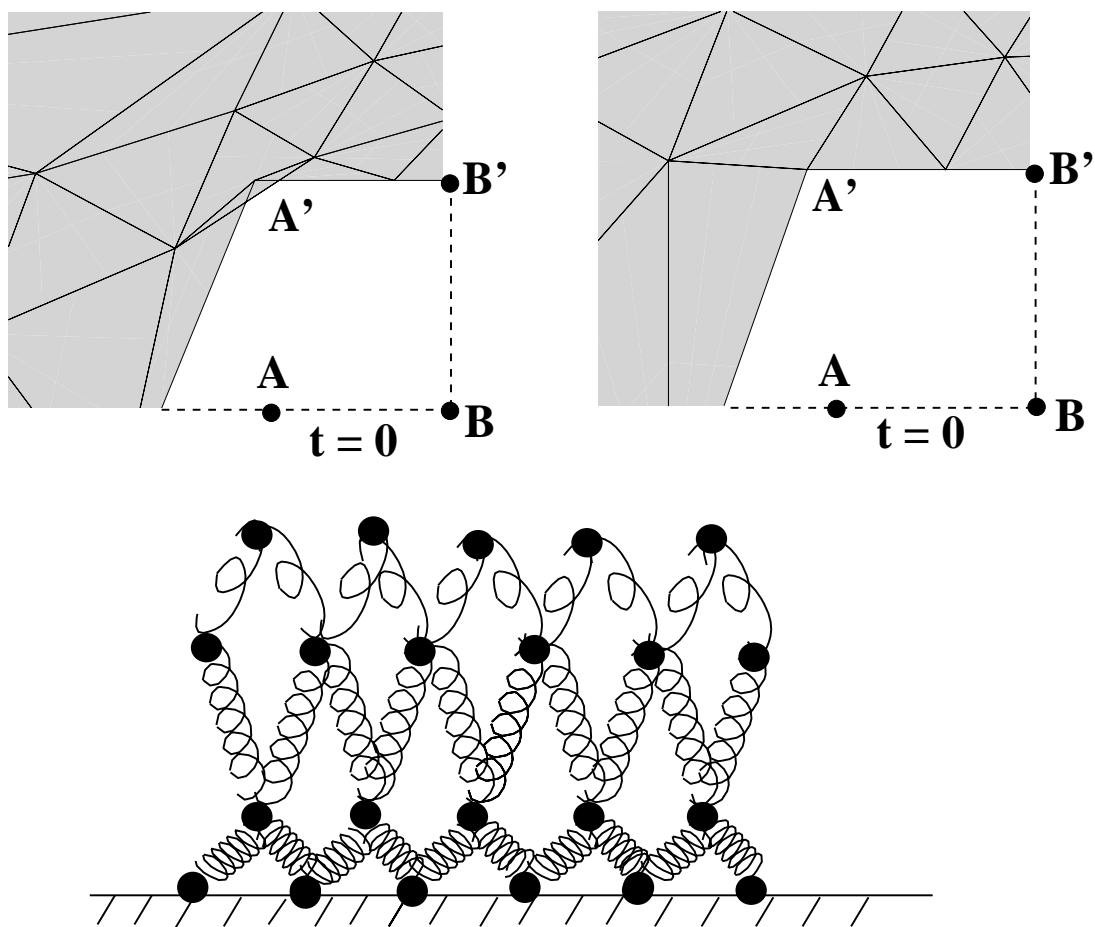
$$u^g = \sum_{i=1}^{deg(v)} \alpha_i^l u_i, \quad \sum_{i=1}^{deg(v)} \alpha_i^l = 1,$$

where  $deg(v)$  is the number of edges meeting at the vertex  $v$  and  $\alpha_i^l$  is the  $l$ th layer weight associated with the  $i$ th edge. This is subjected to the following constraints:  $u^g = 0$  ( $\forall v \in V_1$ ), and  $u^g$  ( $\forall v \in V_0$ ) is prescribed to be the wall velocity. This procedure is repeated for a few cycles following an *incomplete* iteration algorithm, over all  $v \in V_2$ . (Here by incomplete we mean that only a few sweeps are performed and not full convergence is sought.) Once the grid velocity is known at every vertex, the updated vertex positions are determined using explicit time-integration of the newly found grid velocities. An example is shown in figure 13.

In summary, the following are the main features of the ALE-Graph formulation:

- It introduces an “arbitrary” mesh velocity into the variational formulation for the governing equations.
- It reduces to a Lagrangian description at the moving structure and Eulerian description at the stationary regions.

# The ALE Grid Velocity Algorithm



Graph Theory: Force Directed Method in Velocity Space

Fast Analog of  $\nabla \cdot (\kappa \nabla U_g) = 0$  Using Incomplete Iteration

Figure 13: Variable tension layering for the graph edges (bottom) and result of the classical smoothing (top left) versus the graph-based velocity grid algorithm (top right).



- It computes a mesh velocity for the vertices only (not the interior points of a spectral element) using a variable tension/stiffness force-directed graph-based method that is efficient.

## 4 Hardware Comparison

Serial and parallel algorithms for unstructured grids were tested based on the  $\mathcal{N}\epsilon\kappa\mathcal{T}\alpha r$  code. For the serial algorithms, two-dimensional spectral/ $hp$  element simulations were run to evaluate the single node performance of the various platforms. Parallel machine capability was tested through the use of the  $21/2D$   $\mathcal{N}\epsilon\kappa\mathcal{T}\alpha r$ -F code. We have limited our study to using at most four processors so that we could include comparisons with a local 4 PC Linux Cluster. Due to this choice, our parallel algorithm based on geometric domain decomposition for fully three-dimensional inhomogeneous flows was not tested because it is not applicable to use four processors for fully 3d DNS since in general four processors cannot accommodate meshes currently used for turbulence simulation of realistic domains.

Comparisons between seven machines are made, namely:

- Cray T3E-900, composed of 816 processor nodes. Each processor runs at 450 MHz and total available memory is 258 GB.
- IBM SP, composed of 211 RS/6000 P2SC nodes. Each node has a single 160 MHz with 256 Mb of memory per node. All of the nodes are connected via a high-speed switch.
- IBM SP2, composed of 24 Thin 2 Nodes. Each node has a single 66MHz Power2 processor and 128Mb of memory.
- Silicon Graphics Onyx2 Infinite Reality, with 2GB of memory and 4 195MHz R10000 processors.
- Fujitsu AP3000 with 28 UltraSPARC 300MHz single processor nodes, each having 256 MBytes per node, connected via a dedicated high-speed network (AP-Net).
- IBM SP with 24 RS/6000 Model F50 (Silver) nodes, with 1GB memory per node and four 604e Power PC 332 MHz processors per node.
- Cluster of 4 PCs, each PC with an Intel 450MHz Pentium II processor. Each PC has 384MB 100MHz synchronous DRAM, 512K pipelined burst cache and an Ultra ATA hard drive. The network connection uses the Adaptec Quad 10/100TX PCI ANA-6944A, Pentium II optimised. The Mandrake Linux 5.3 distribution, upgraded to the 2.2.5 kernel (and associated modules) is used.

## DNS/Serial algorithms

In order to evaluate the single node performance of *NekTar* with serial algorithms, two-dimensional simulations are performed. For the bluff body flow simulation benchmark, 60,000 degrees of freedom were used. Maximum CPU times per time step are shown in Table 1, calculated with the call, `clock()`.

Table 1: CPU time for serial algorithm bluff body flow simulation.

Machine	CPU time (seconds) / time step
Fujitsu AP3000	1.25
SGI Onyx 2	1.04
PC	0.81
SP2 "Thin2" nodes	1.45
SP "Silver" nodes	1.29
SP "P2SC" nodes	0.71
T3E	0.82

From table 1 we see that the P2SC node outperforms all other platforms tested for single CPU performance, being over 10% faster than the nearest competitor, and almost twice as fast as the slowest machine tested (SP2 Thin 2). In second place, we see that both the Pentium II and Cray T3E perform well, both being 20% faster than the next competitor, the Onyx 2. It is interesting to note that both the PC and T3E are running at 450 Mhz, hence with this particular test case we see what appears to be a Mhz to Mhz comparison between the two machines for single CPU performance. **In more recent comparisons for a larger 2D similar problem involving 330,000 degrees of freedom we obtained the following timings per time step:**

- Athlon (600 MHz): 1.2 sec
- Power3 (200 MHz): 1.2 sec
- Compaq 2164 (750 MHz): 0.63 sec
- T3E (450 MHz): 1.78 sec
- Pentium II (450 MHz): 1.74 sec.

## DNS/Parallel algorithms

Parallel tests were run using *NekTar*-F. Each processor computes a certain number of Fourier modes. Typically, one processor is assigned to one Fourier mode which corresponds to two physical degrees of freedom, defining two spectral/hp element planes. Therefore, if four Fourier modes are used for resolving the homogeneous directions, then there are eight spectral/hp element planes. Increasing the resolution

in the homogeneous direction, i.e. increasing the number of Fourier planes can be accomplished without increasing the amount of memory required per processor if the number of processors is increased. We present wall clock times for a four CPU parallel simulation having 460,000 degrees of freedom.

Table 2: Wall Clock time for 4 Processor parallel algorithm applied to a bluff body flow simulation.

Machine	Wall Clock time (seconds) / time step
Fujitsu AP3000	4.45
SGI Onyx 2	4.64
PC	6.20
SP2 “Thin2” nodes	6.07
SP “Silver” nodes	5.73
SP “P2SC” nodes	3.11
T3E	3.02

For the code used to obtain the numbers presented in table 2, the main types of communication patterns used are: Global Exchanges (All-to-All), Global Reduction/Gather operations (addition, min, max) and Global Synchronization. Hence our analysis compares parallel performance based on machine wide operations.

For parallel performance, the T3E and IBM P2SC compare favorably with each other when comparing four CPU wall clock times. These two machines are reasonably ahead of all other platforms tested. Comparing the IBM SP P2SC nodes with the slowest machines, the PC Cluster and the IBM SP2 Thin-2 nodes, we find that the P2SC nodes and T3E nodes give us almost a factor of two speedup.

## Scalability

Due to the algorithmic design of the *NekTar*-F code, we achieve almost perfect problem size scalability. In the case that more Fourier planes are needed for resolution in the homogeneous direction, additional processors can be used with little overhead. In addition to the results presented above, we have obtained performance results for larger numbers of processors for a few specific cases. When comparing the P2SC nodes to the Thin-2 nodes on 16 processor jobs, we find that the P2SC nodes continue to give us a speedup of two based on wall clock time. Comparing the T3E-900 against the 195 MHz Origin 2000, we find that for 88 processor ALE simulations the T3E provides us with approximately a 20% speedup.

A very recent result from simulations of turbulent wakes formed in flow past a cylinder is shown in figure 14. All simulations were performed at MHPCC under dedicated computer time. There are about 20,000 dofs per Fourier plane and the total number of dofs increase linearly with the number of processors, so here we test the scaled parallel efficiency of the system.

## Acknowledgments

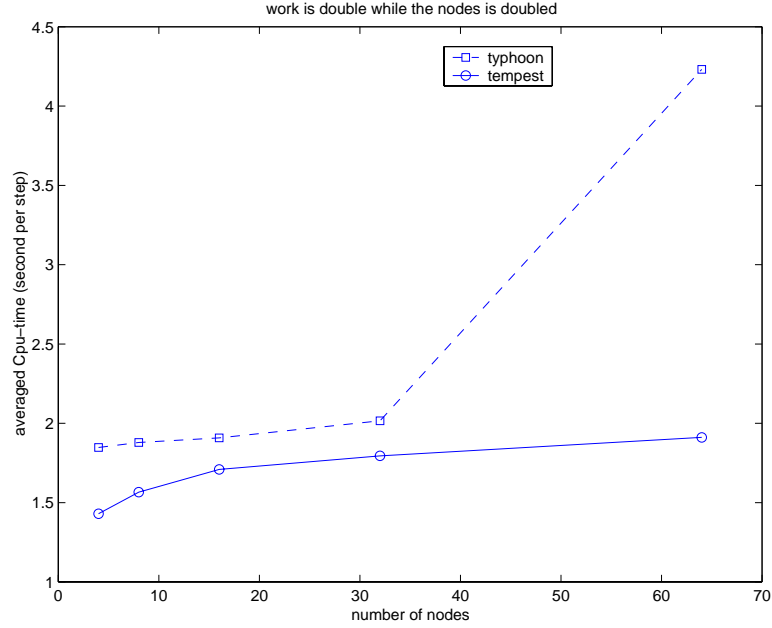


Figure 14: Comparison of the two systems at MHPCC for 3D flow past a cylinder ( $\mathcal{N}\epsilon\kappa\mathcal{T}\alpha r$ -F). In the plot, “typhoon” refers to the P2SC-based IBM SP system and “tempest” refers to the Power3-based IBM SP. The problem size scales up as the number of processors increase so here we compare the scaled efficiency of the two systems.

This work was supported by ONR, AFOSR and DARPA, and computations were done on the IBM SP2 at Maui High Performance Computing Center and on the T3E-900 at NAVO.

## References

- [1] Y. Du and G.E. Karniadakis. Suppressing wall turbulence by means of a transverse traveling wave. *Science*, 288:1230, 2000.
- [2] G.E. Karniadakis and S.J. Sherwin. *Spectral/hp Element Methods for CFD*. Oxford University Press, 1999.
- [3] S.J. Sherwin and G.E. Karniadakis. Tetrahedral *hp* finite elements: Algorithms and flow simulations. *Journal of Computational Physics*, 122:191, 1995.
- [4] T.C. Warburton and G.E. Karniadakis. A discontinuous Galerkin method for the viscous MHD equations. *Journal of Computational Physics*, 152:608–641, 1999.
- [5] B. Cockburn, G.E. Karniadakis, and C.-W. Shu. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer, Lecture Notes in Computational Science and Engineering, 2000.
- [6] M. Gad el Hak. *Flow Control: Passive, Active and Reactive Flow Management*. Cambridge University Press, London, 2000.

- [7] A.A. Townsend. *The Structure of Turbulent Shear Flow*. Cambridge University Press, 1976.
- [8] C. Evangelinos and G.E. Karniadakis. Dynamics and flow structures in the turbulent wake of rigid and flexible cylinders subject to vortex-induced vibrations. *J. Fluids Mech.*, 400:91–124, 1999.
- [9] C. Evangelinos, I. Lomtev, and G.E. Karniadakis. Shedding patterns in flow-structure interactions. *J. Visualization*, 2(2):135–142, 1999.
- [10] G.E. Karniadakis. Simulating turbulence in complex geometries. *Fluid Dynamics Research*, 24(6):343–362, 1999.
- [11] G.S. Karamanos, C. Evangelinos, R. Boes, R.M. Kirby, and G.E. Karniadakis. Dns of turbulence on a pc/linux cluster: Fact or fiction? *Proc. SuperComputing 1999*, 1999.
- [12] R.M. Kirby, T.C. Warburton, A. Beskok, and G.E. Karniadakis. ‘the nektar code: Dynamic simulations without remeshing. *Proc. 2nd International Conference on Computational Technologies for Fluid/Thermal/Chemical Systems with Industrial Applications*, August 1-5, 1999.
- [13] X. Ma, G.-S. Karamanos, and G.E. Karniadakis. Dynamics and low-dimensionality of turbulent near-wake. *J. Fluid Mech.*, 410:29–65, 2000.
- [14] C. Evangelinos, D. Lucor, and G.E. Karniadakis. Dns-derived force distribution on flexible cylinders subject to vortex-induced vibration. *J. Fluids & Structures*, 14(3):429–440, 2000.
- [15] G.-S. Karamanos and G.E. Karniadakis. A spectral vanishing viscosity method for LES. *J. Comp. Phys.*, to appear, 2000.
- [16] Ma Xia, G.E. Karniadakis, G.S. Karamanos, and S.J. Sherwin. Issues in les of wake flows. *AIAA 98-2893, 29th AIAA Fluid Dynamics Conference, Albuquerque, NM*, June 15-18, 1998.
- [17] Ma Xia and G.E. Karniadakis. The spectrum of the turbulent near-wake: A comparison of dns and les. *1st AFOSR Int. Conference on DNS/LES, Ruston, LA*, 1997.
- [18] C. Evangelinos and G.E. Karniadakis. Correlation length and forces in flow past a freely-oscillating cylinder. *Proc. ASME Fluids Engineering Division Summer Meeting*, eds. P. Bearman and C.H.K. Williamson, Washington DC, June 21-25, 1998.
- [19] E. Tadmor. Convergence of spectral methods for nonlinear conservation laws. *SIAM J. Numer. Anal.*, 26(1):30, 1989.
- [20] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph Drawing*. Prentice Hall, 1998.
- [21] R. Lohner and C. Yang. Improved ale mesh velocities for moving bodies. *Comm. Num. Meth. Eng. Phys.*, 12:599–608, 1996.